Modeling Basic Maneuvering with the Predator UAV

11 Jan 02

Kevin Gluck
Air Force Research Laboratory

Jerry Ball
Mike Krusmark
L-3 Communications

AFOSR Grant #: 02HE01COR

Outline

- Background and Model Design (Kevin)
- Model Development and Implementation (Jerry)
- Model Assessment (Mike)
- Model Achievements, Shortcomings, and Future Directions (Kevin)

Mission

Research, develop, demonstrate, evaluate, and transition leading edge training technologies and methods to train warfighters

Challenge

More constraints on live training …

(*ranges, budgets, equipment wear, ops tempo, safety*)

… yet modern weapons and employment concepts require extensive airspaces, large ranges, and more composite training opportunities.

Distributed Mission Training

DMT is likely to play an increasingly important role in future warfighter training.

Representation of human performance and learning is one of the great challenges to overcome before the full potential of DMT can be realized.

Cognitive Modeling Applications

Better human behavior representation can improve the effectiveness and flexibility of DMT

- **Model Warfighter Behavior**

  (Computer-Generated Forces)

- **Predict Warfighter Behavior**

  (Automated Training Program Assessment)

- **Assess Warfighter Behavior**

  (Instructional Agents)

<u>Domains:</u> Air, C2, Space, IW . . .

Performance and Learning Models Research Program

Objective

Advance the state of the art in computational process modeling of human-system interaction in dynamic, time-constrained environments.

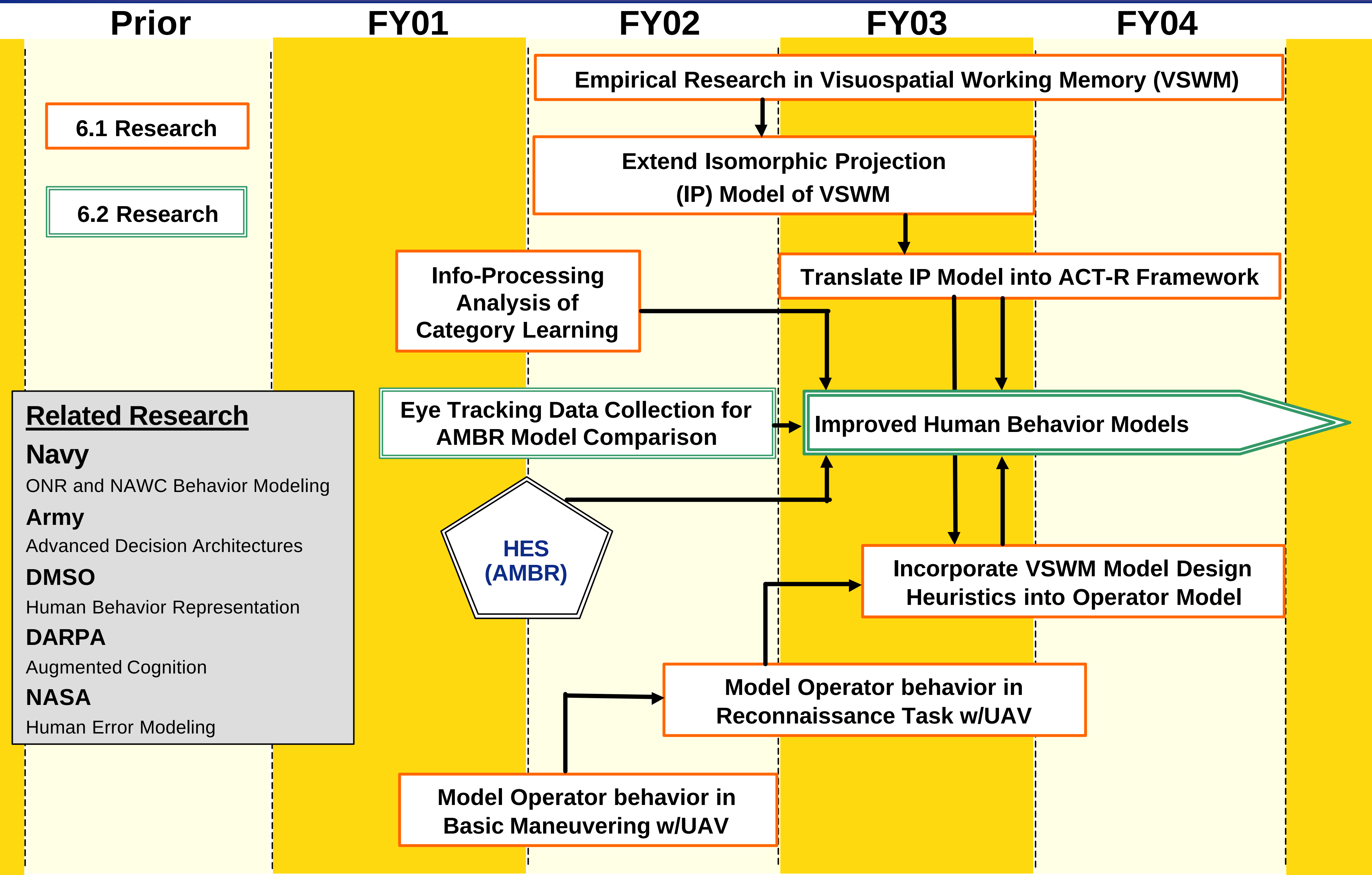
Goals

- Develop computational process models representing the behavior of UAV operators.
- Use those models to explore the role of visuospatial working memory in determining operator performance in a realistic UAV reconnaissance task.
- Transition products and lessons learned both to the scientific community and to applied research involving warfighter modeling.

UAV Synthetic Task Environment

- Developed with New World Vistas funding from AFOSR
- Includes basic maneuvering, reconnaissance, and landing tasks
- Dynamic environment involving time-constrained decision-making and psychomotor skill

Roadmap



Prior
FY01
FY02
FY03
FY04
6.1 Research
6.2 Research
Empirical Research in Visuospatial Working Memory (VSWM)
Extend Isomorphic Projection (IP) Model of VSWM
Info-Processing Analysis of Category Learning
Translate IP Model into ACT-R Framework
Eye Tracking Data Collection for AMBR Model Comparison
Improved Human Behavior Models
HES (AMBR)
Incorporate VSWM Model Design Heuristics into Operator Model
Model Operator behavior in Reconnaissance Task w/UAV
Model Operator behavior in Basic Maneuvering w/UAV
Related Research
Navy
ONR and NAWC Behavior Modeling
Army
Advanced Decision Architectures
DMSO
Human Behavior Representation
DARPA
Augmented Cognition
NASA
Human Error Modeling

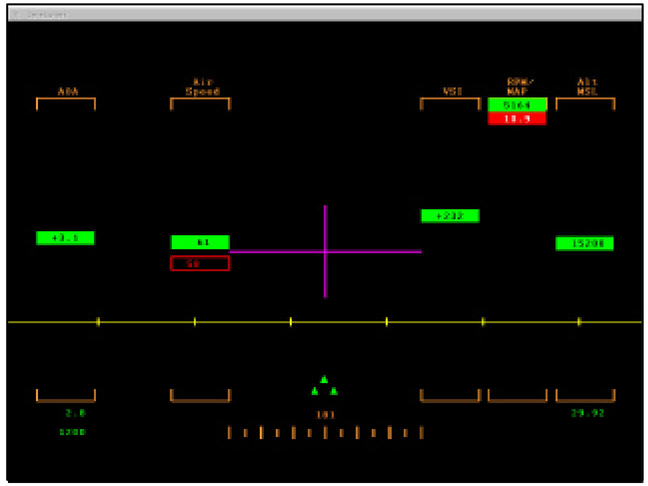
UAV Operator Model Plan

Step 1

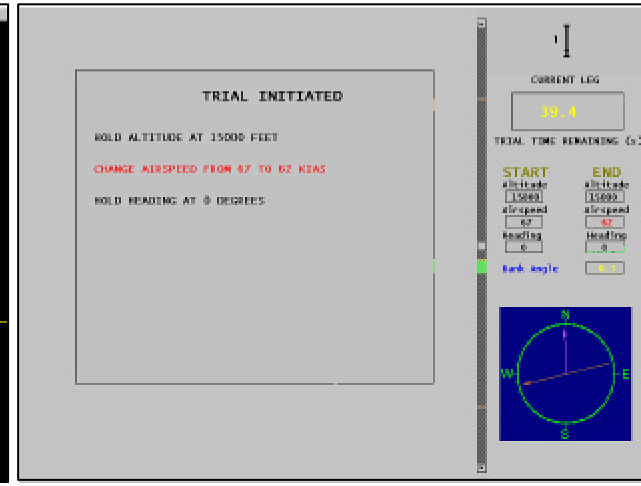
Basic Maneuvering

(We are here)

Heads-Up Display



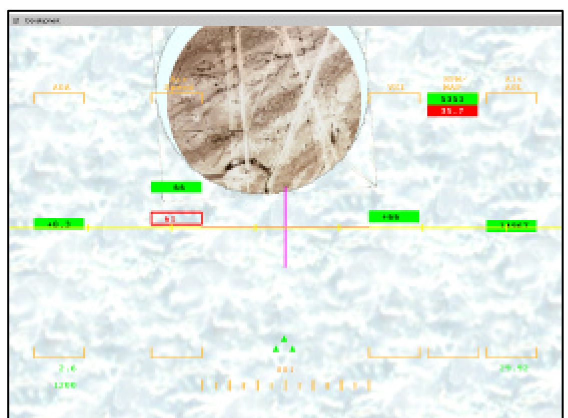
Task Screen



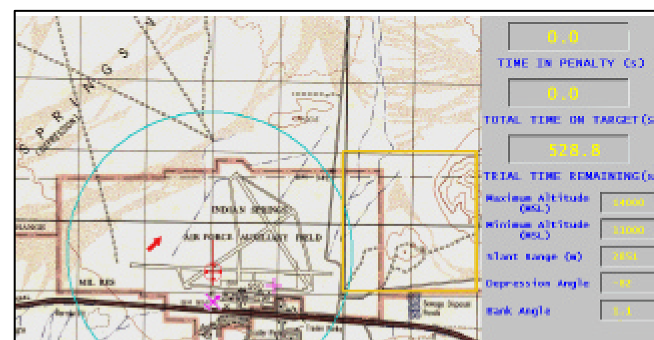
Step 2

Reconnaissance

Ground Camera



Tracker Map



Basic Maneuvering

Segment 1

Hold altitude at 15000 feet.

Change airspeed from 67 to 62 knots.

Hold heading at 0 degrees.

Model does Segment 1.

Segment 2

Hold altitude at 15000 feet.

Hold airspeed at 64 knots.

Change heading from 0 degrees to 180 degrees.

-

-

-

Segment 7

Change everything!

Sources of Data

- Basic Maneuvering Tutorial and Segment Instructions
- Local aviation expertise
  - LtCol Stu Rodgers

Primary influences on model implementation to date.

- Subject Matter Expert
  - Verbal protocols
  - Eye movements
  - Data files

Recently collected. Under-utilized to date.

- Active-Duty AVO Computer Log Files
  - Control Inputs
  - Performance Deviation Data

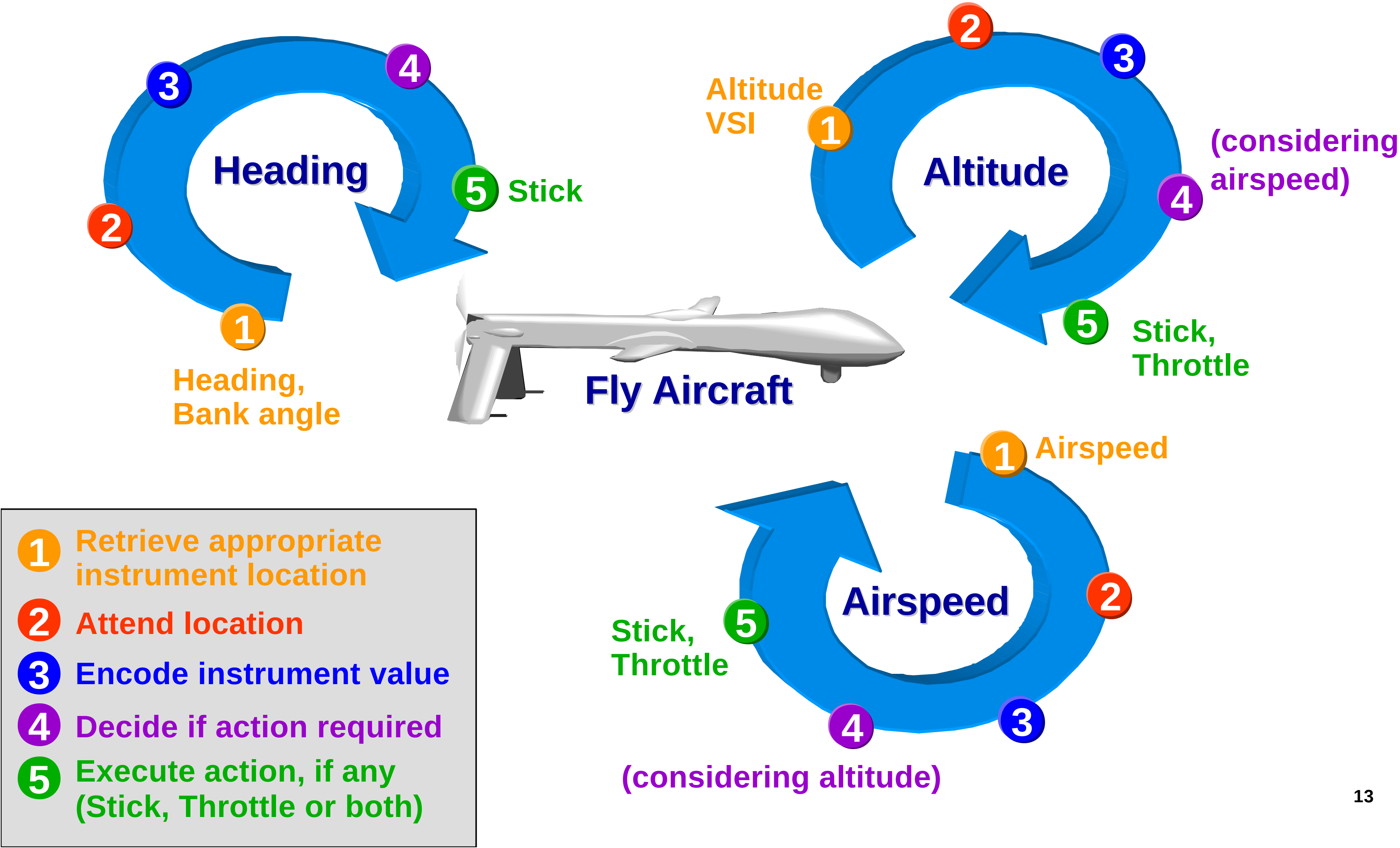
Model assessment. (Later in presentation)

Motivation for Model Design

- “Unit Task” representation
  - Newell
  - Kanfer-Ackerman ATC Model (Lee and Anderson)
  - AMBR ATC Model (Lebiere, Anderson, and Bothell)

Model Design

Three unit tasks for performance monitoring (heading, airspeed, altitude)

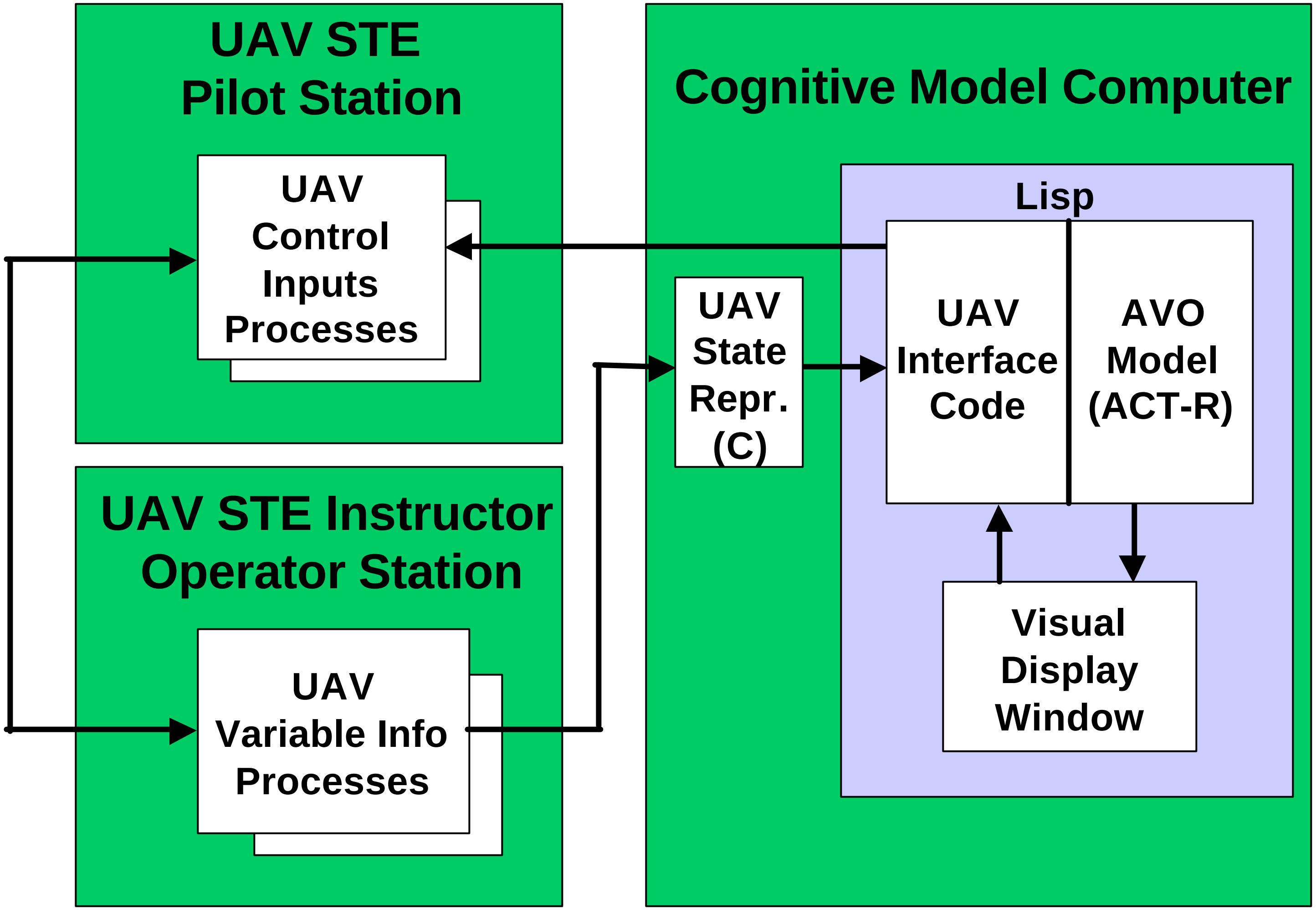


Model Development and Implementation

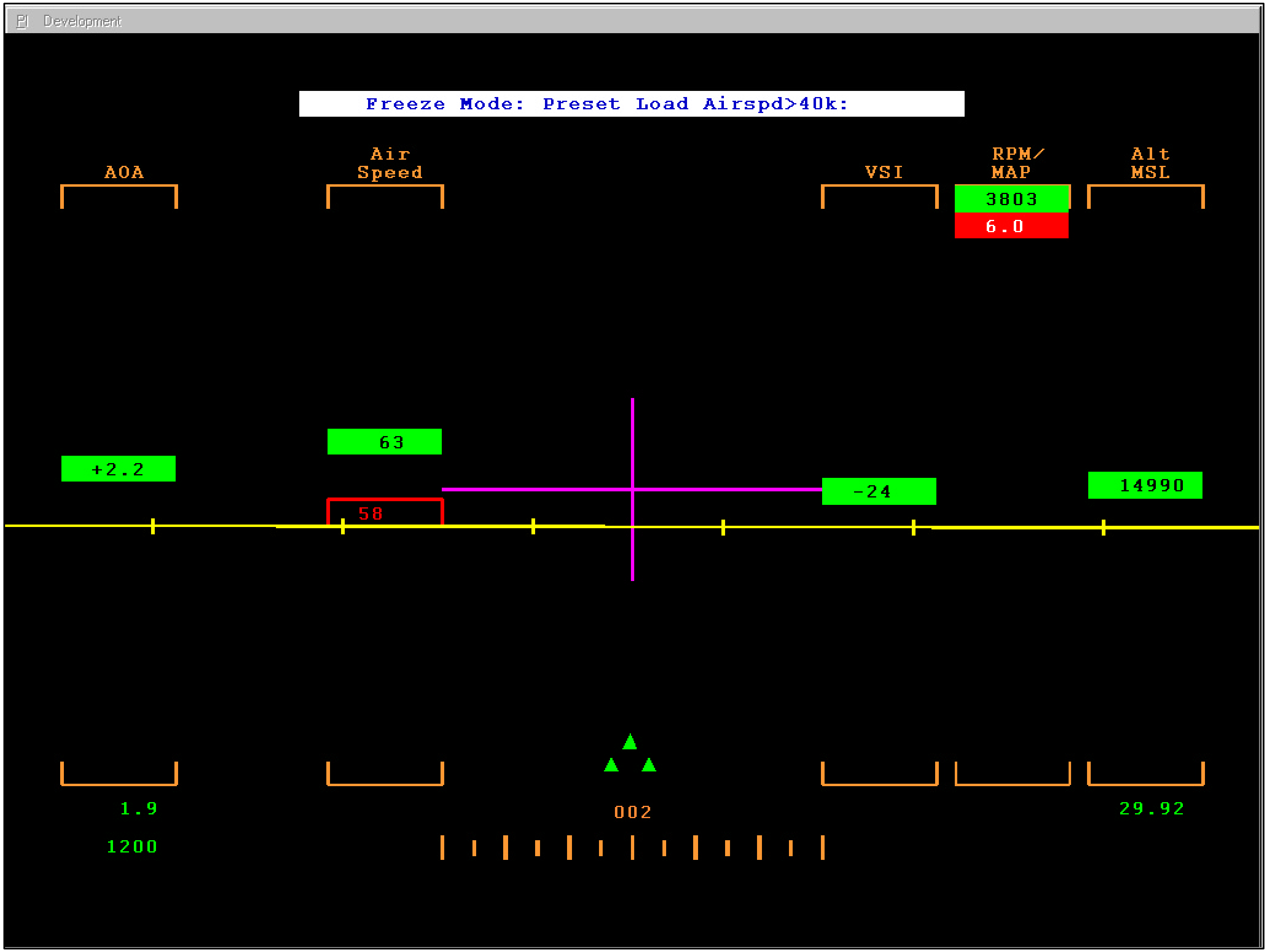
Model Development

- System Architecture
- Interface to UAV STE
- Development Environment
- Model Implementation

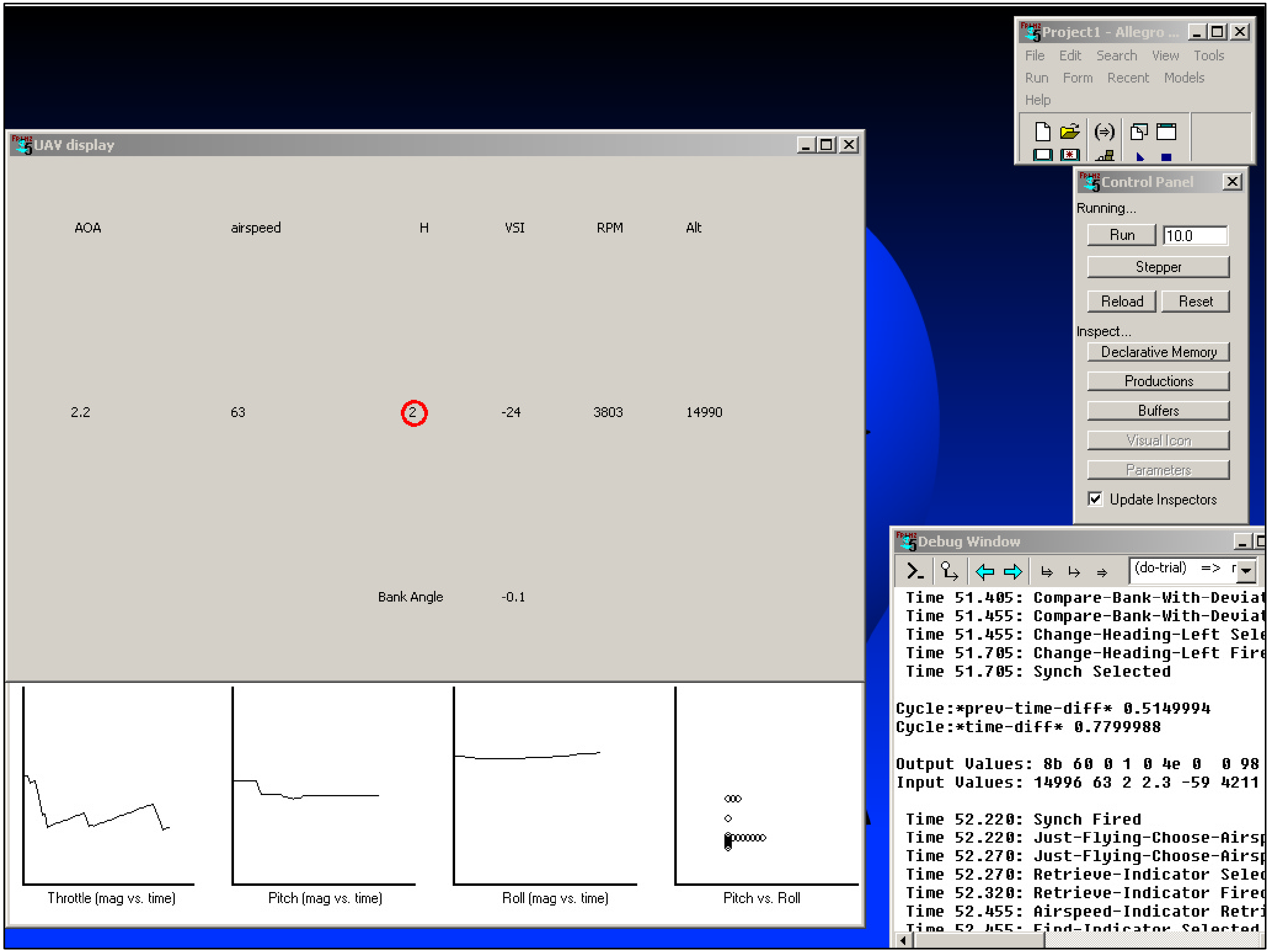
System Architecture



UAV Heads-Up Display



ACT-R Heads-Up Display



Interface to UAV STE

- UAV STE consists of 11 separate processes on 2 machines using proprietary interface to communicate
- No Access to Source Code for “Heartbeat” Mechanism of STE. Cognitive Model had to be synch’ed to STE
- Cognitive Model runs SLOWER than STE with graphics for HUD Mockup running
- Shared Memory between Cognitive Model and C Process on same machine used to receive Variable Data from STE
- Windows Event Processing Loop in STE modified to receive Datagram Events from Cognitive Model for Control Inputs (circumventing Control Inputs from Stick and Throttle)

Development Environment

- Change from ACT-R 4.0 to 5.0 under Allegro CL 5.0.1
- Looked at using Java version of ACT-R
  - determined that it's not ready for prime time
- Re-implementation of Heads-Up Device (HUD) Mockup in Cognitive Model in order to use RPM component of ACT-R 5.0
- No Hands on Throttle and Stick (HOTAS) support in ACT-R 5.0

Model Implementation

- Goal Chunk Type
- Select Unit Task Productions (Random)
- Retrieve Indicator Location Production
- Find Indicator Production
- Attend Indicator Production
- Encode Indicator Value Production
- Decide On Action Productions
- Execute Action Productions

Standard Visual Attention Representation

Goal Chunk Type

```
(chunk-type fly-aircraft
   state
   indicator
   current-value
   current-airspeed
   desired-airspeed
   previous-airspeed
   max-desired-airspeed
   min-desired-airspeed
   current-altitude
   desired-altitude
   max-desired-altitude
   min-desired-altitude
   current-heading
   desired-heading
   drifting
   heading-deviation
   current-vertical-speed)
```

Select Unit Task - Altitude

One of three unit task selection productions.

```
(p just-flying-choose-altitude
   =goal>
       ISA fly-aircraft
       state start
 ==>
   =goal>
       state retrieve
       indicator alt-indicator)
```

Chosen randomly.

Retrieve Instrument Location

A single production for retrieving info. about an indicator (any indicator)

```
(p retrieve-indicator
   =goal>
      ISA fly-aircraft
      state retrieve
      indicator =indicator
 ==>
   =goal>
      state find
   +retrieval>
      ISA instrument
      name =indicator)
```

Find Indicator

```
(p find-indicator
   =goal>
      ISA fly-aircraft
      state find
   =retrieval>
      ISA instrument
      location =loc
 ==>
   +visual-location>
      ISA visual-location
      nearest =loc
   =goal>
      state attend)
```

Attend Indicator

```
(p attend-indicator
    =goal>
        ISA fly-aircraft
        state attend
   =visual-location>
        ISA visual-location
   =visual-state>
        ISA module-state
        modality free
==>
  +visual>
        ISA visual-object
        screen-pos =visual-location
  =goal>
        state encode)
```

Encode Indicator

```
(p encode-indicator
    =goal>
       ISA fly-aircraft
       state encode
    =visual>
       ISA text
       value =current-value
 ==>
   !bind! =integer-value (get-value =current-value)
   =goal>
       state compare
       current-value =integer-value)
```

Decide On Action

IF altitude is high and climbing or level
THEN decrease altitude

```
(p compare-high-altitude-and-VSI--decrease-altitude
   =goal>
       ISA fly-aircraft
       indicator vert-speed-indicator
       state compare
       max-desired-altitude =max-desired-altitude
       > current-altitude =max-desired-altitude
       current-value =vertical-speed
       >= current-value 0
 ==>
   =goal>
       state decrease-altitude
       current-vertical-speed =vertical-speed)
```

Execute Action

```
(p high-and-slow-decrease-altitude-stick
  =goal>
      ISA fly-aircraft
      state decrease-altitude
      current-altitude =current-altitude
      current-vertical-speed =current-vertical-speed
      desired-altitude =desired-altitude
      min-desired-airspeed =min-desired-airspeed
      < current-airspeed =min-desired-airspeed
 ==>
 !eval! (stick-action-high-and-slow-decrease-altitude
        'forward =current-altitude =desired-altitude
        =current-vertical-speed)
  =goal>
      state nil)
```

Unit Task Decisions

- Heading Unit Task – compare current heading deviation to current bank angle to decide if action is required
- Airspeed Unit Task – compare current airspeed to desired airspeed and current altitude to desired altitude to decide which actions (if any) to take
- Altitude Unit Task – compared current altitude to desired altitude and current airspeed to desired airspeed to decide which actions (if any) to take

Stick and Throttle Actions

- Heading Unit Task – increment or decrement the stick position in the roll plane
- Airspeed Unit Task – adjust the throttle and/or stick position in the pitch plane based on the difference between the current airspeed and the desired airspeed
- Altitude Unit Task – adjust the throttle and/or stick position in the pitch plane based on the difference between the current altitude and the desired altitude and the deviation of the vertical speed from 0

Model Assessment

Human-Model Comparison Data

- Control Inputs
  - Throttle
  - Pitch
  - Roll
- Performance Deviation Data
  - Altitude
  - Airspeed
  - Heading

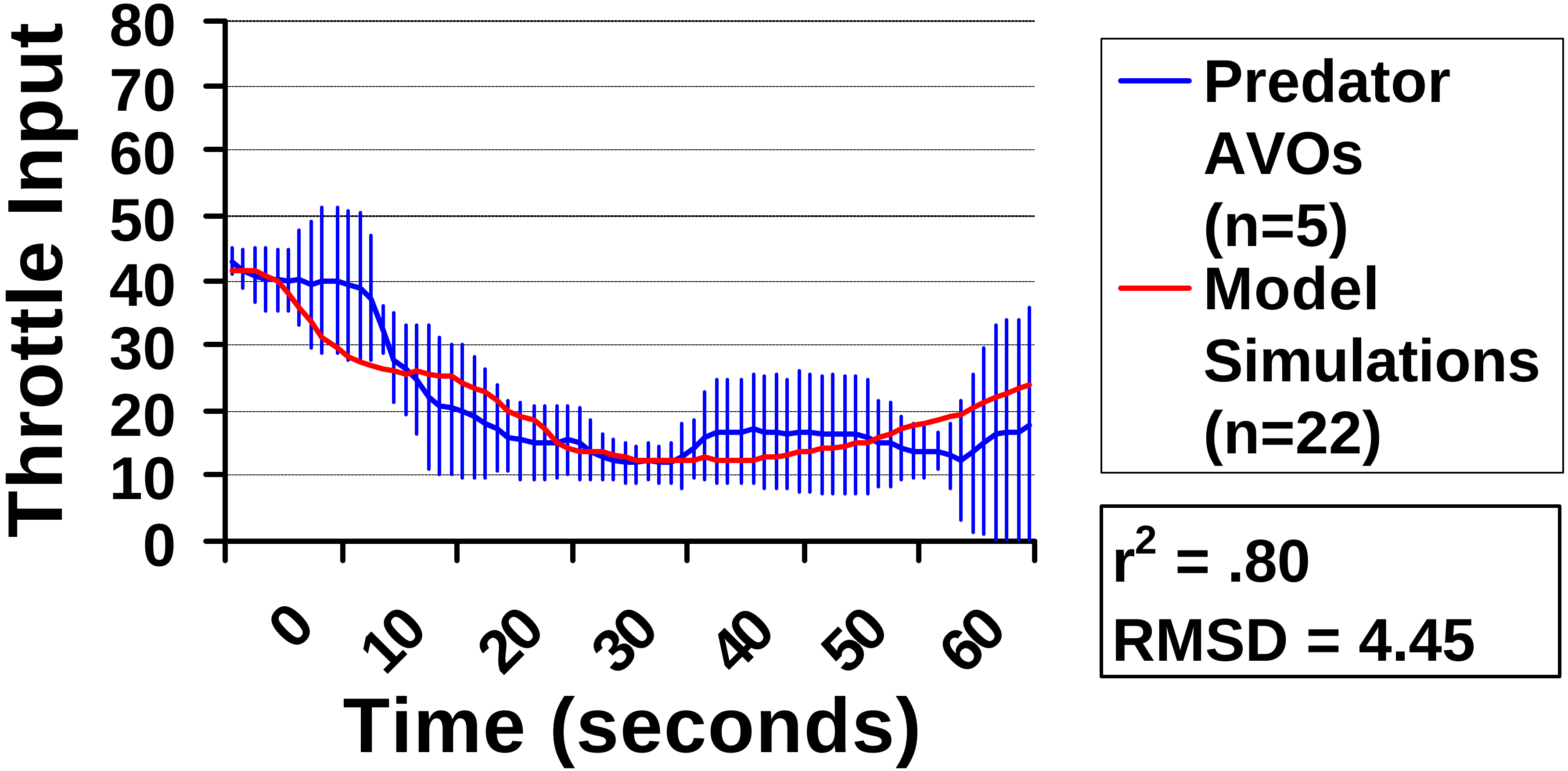
Data are from successful trials only

– when pilot/model meets all three performance criteria.

Presentation of human and model data inspired by Schunn and Wallach (submitted for publication).

Throttle Inputs

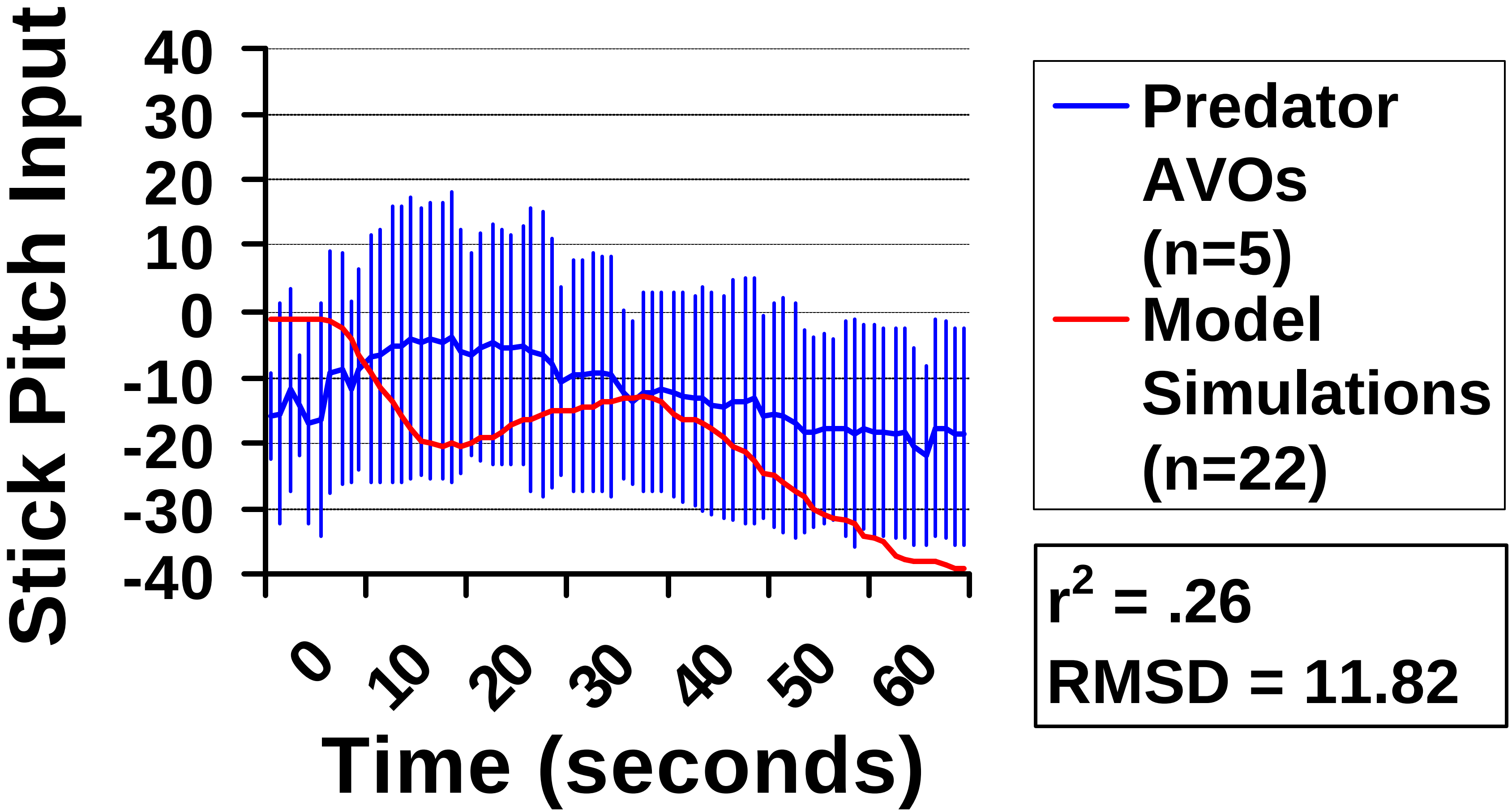
Data-Model Comparison



r^2 = .80

RMSD = 4.45

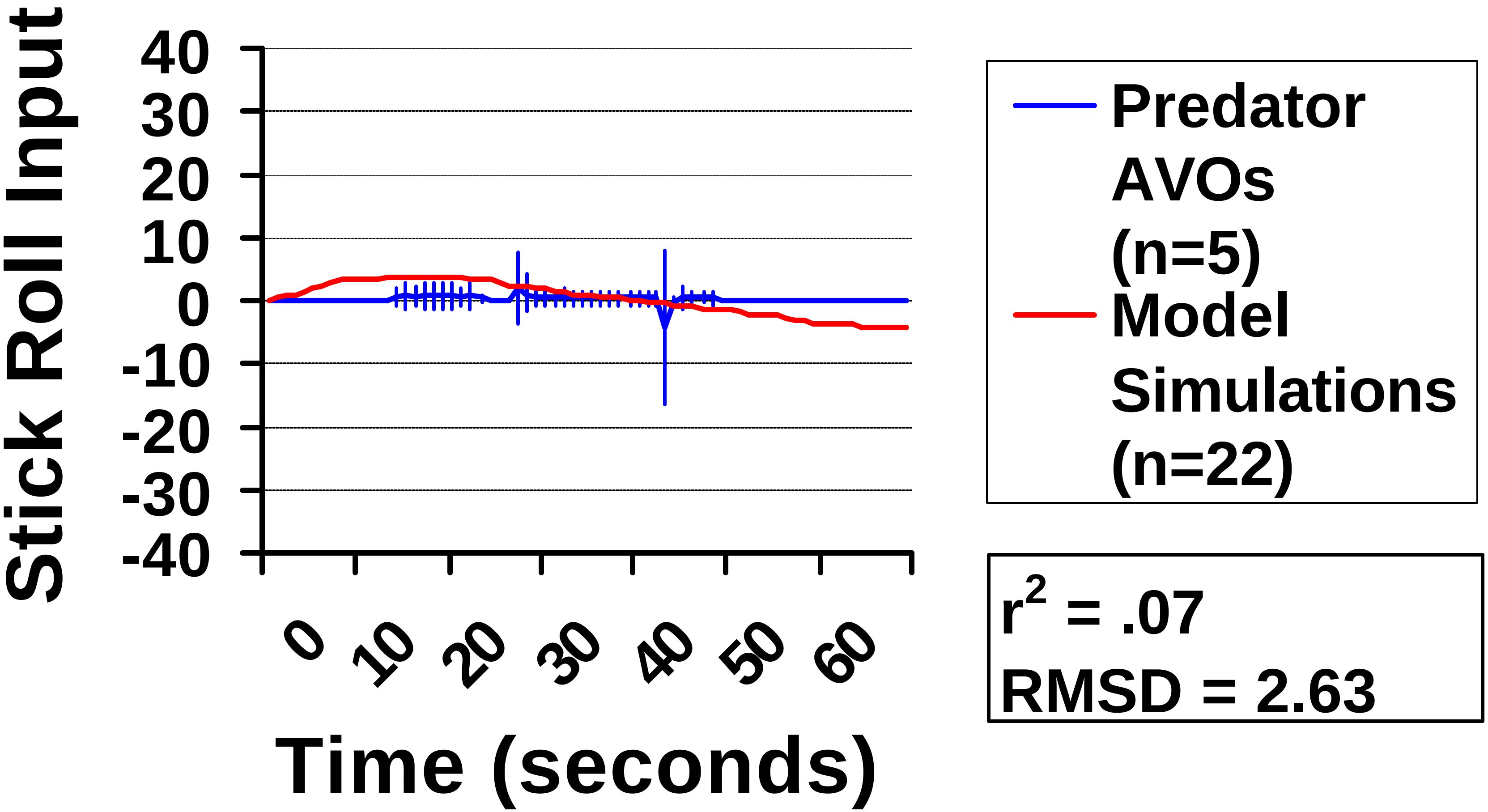
Pitch Inputs
Data-Model Comparison



$r^2 = .26$

RMSD = 11.82

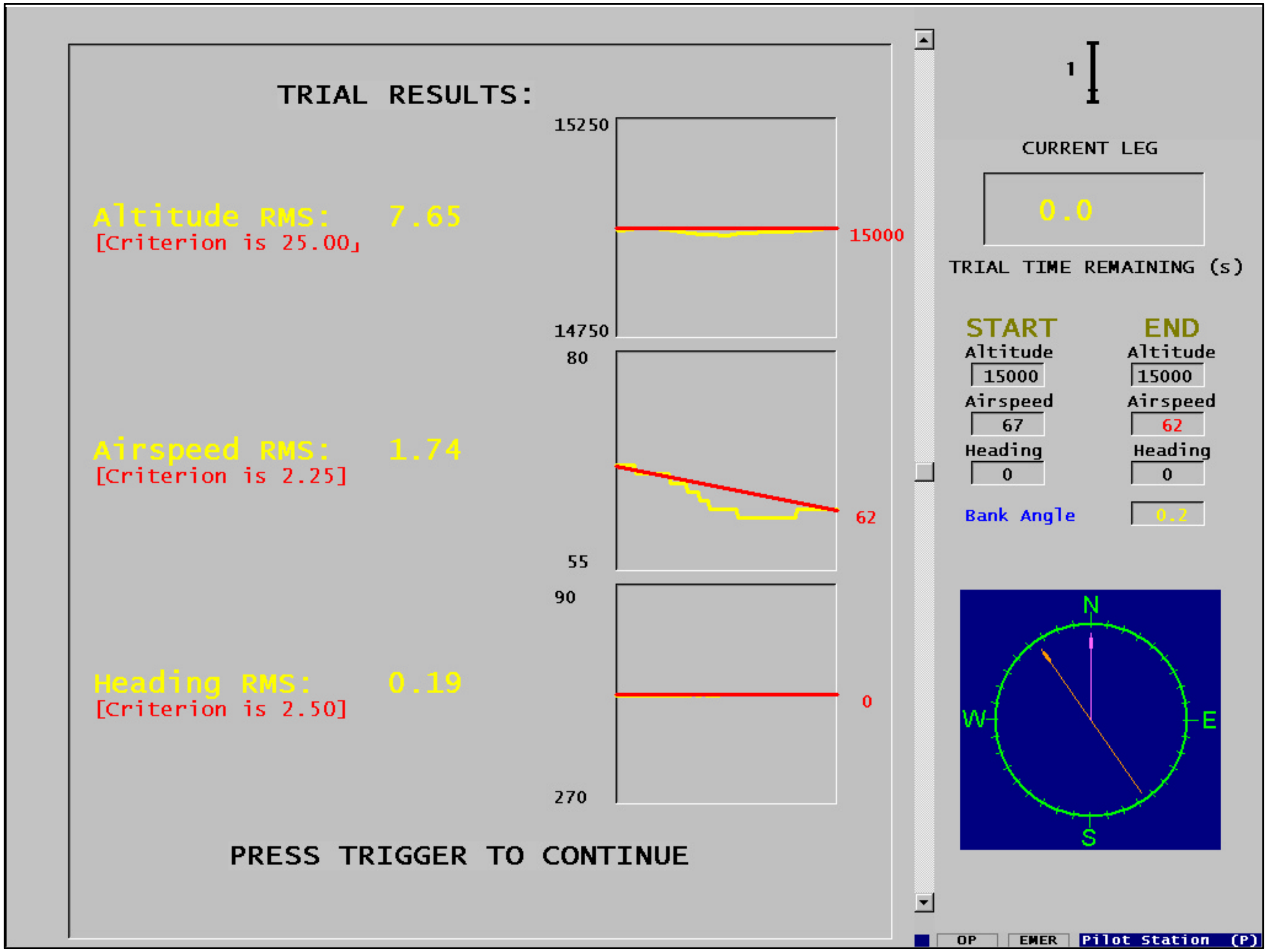
Roll Inputs
Data-Model Comparison



$r^2 = .07$
RMSD = 2.63

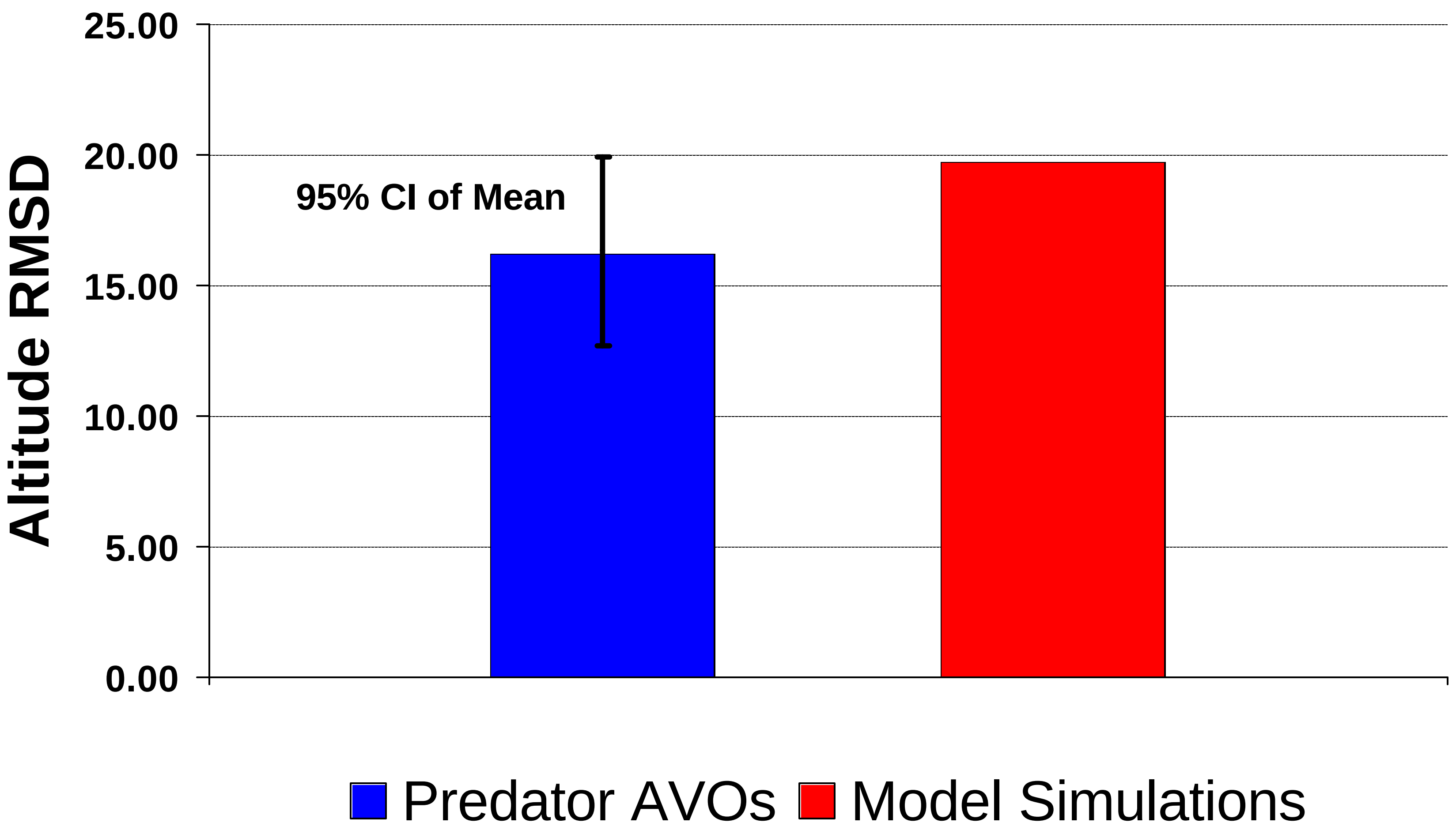
Performance Deviation Data

Feedback Screen

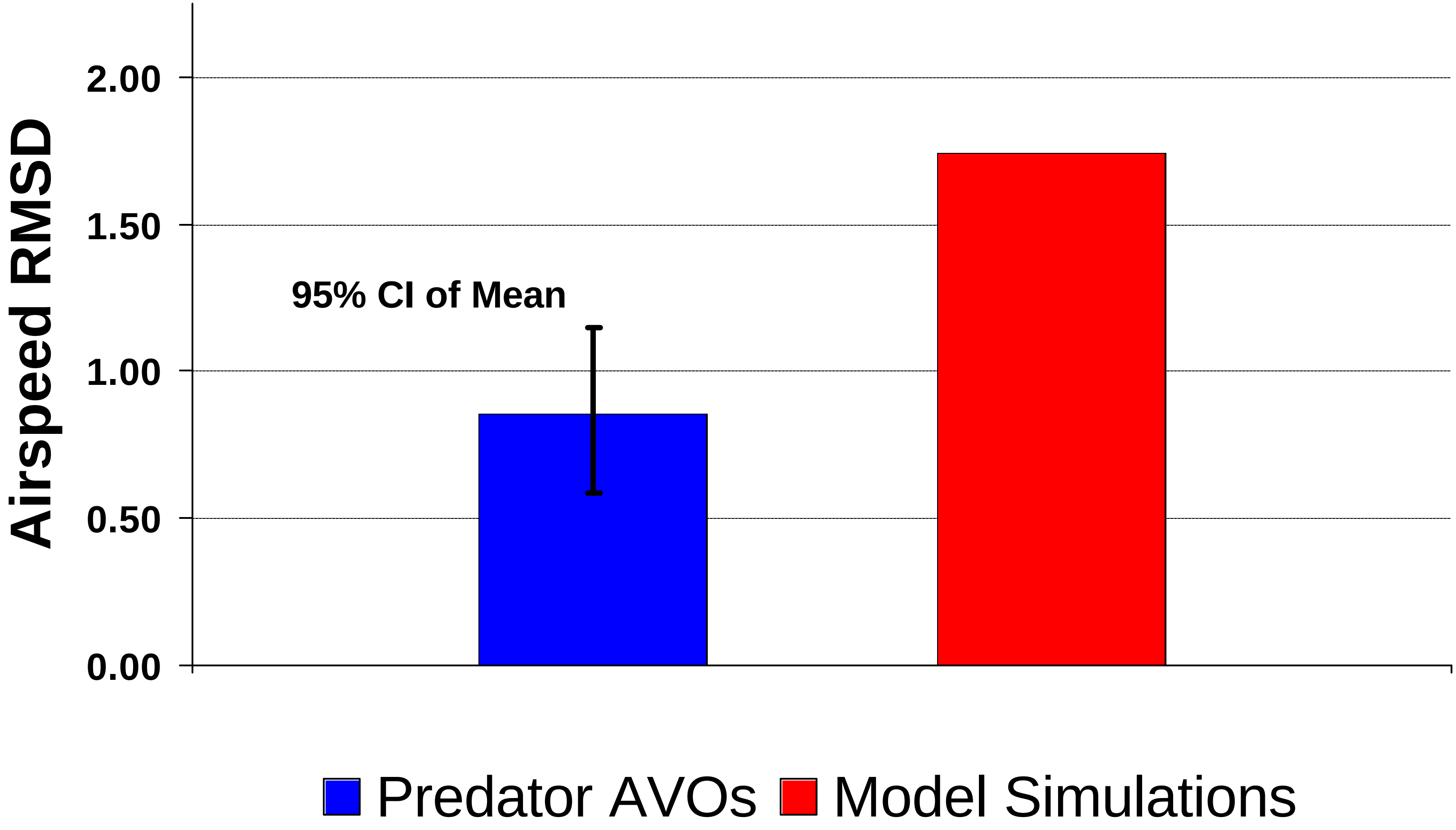


= ideal performance = actual performance RMS = Root Mean Squared Deviation

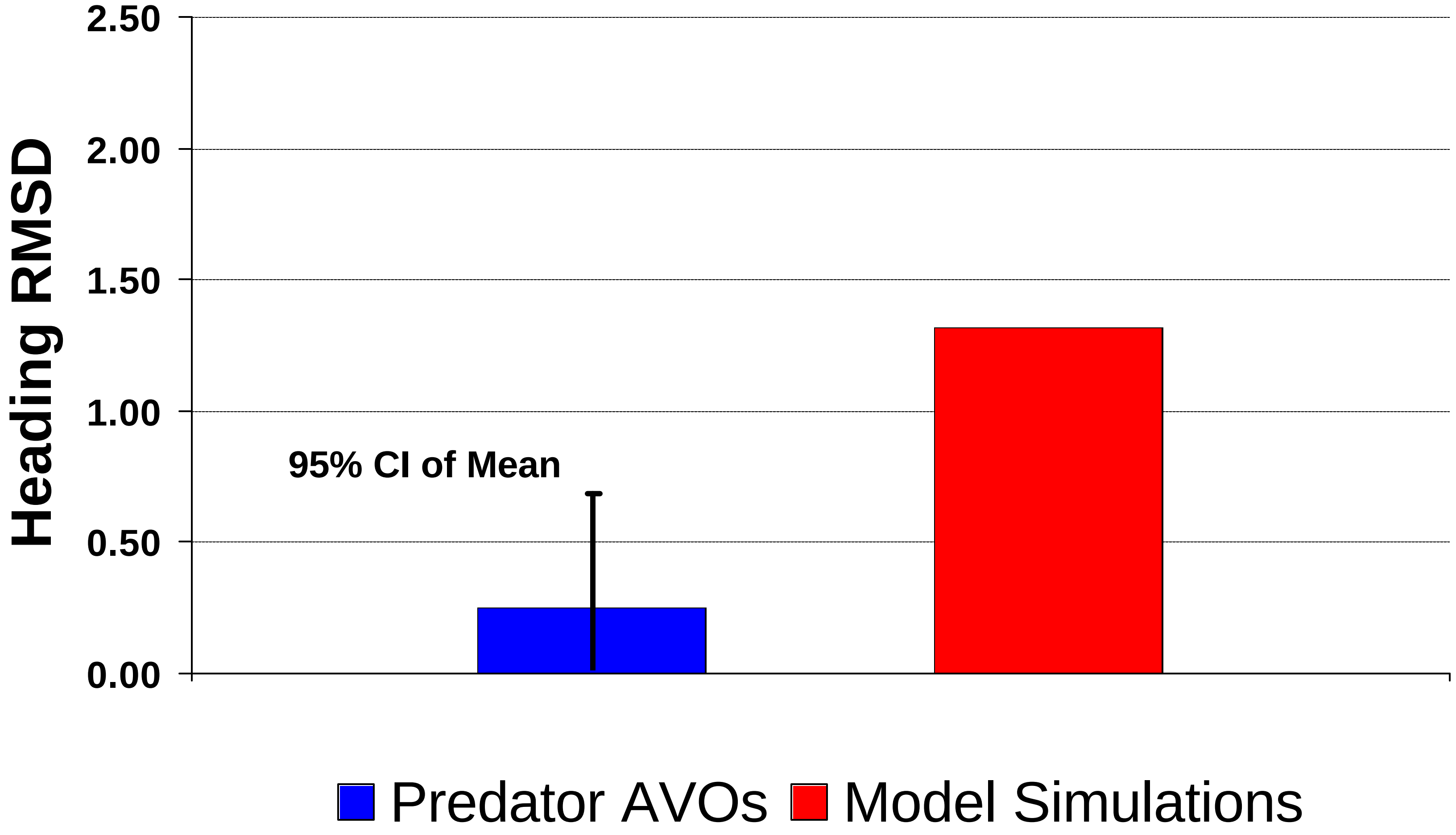
Average of Altitude RMSD’s



Average of Airspeed RMSD's



Average of Heading RMSD's



Wrap-Up

Model Achievements

Linked ACT-R directly to a fairly complex, high-fidelity simulation environment that was *not* originally designed for interaction with cognitive models.

Established baseline level of performance for initial model (Version 1.0), against which future improvements can be measured.

- **Quantitative fit to control inputs**
- **Passes 40% of attempts at basic maneuvering segment 1**

Model Shortcomings
(areas for improvement)

- Magnitude of control inputs is equation-based
  - Not learnable, circumvents the architecture
  - Go to declarative representation
  - Borrow from Lebiere & Wallach (in press) where possible
- Unit tasks are selected randomly from conflict set
  - Can't adjust to short-term task demands (prospective goal setting)
  - Implement cross-checking via declarative retrieval
- No use of horizon line and reticle
  - Clearly wrong, must add
- Attention to clock (and other awareness of time)
- Anticipation of future state; awareness of "getting close"
  - (roll-outs)
- Awareness of "lead-in" period
- Does not learn (yet)

Future Research

Make more use of eye tracking data and verbal protocols

Extend model to remaining basic maneuvering segments

Extend model to reconnaissance missions

Incorporate VSWM model design heuristics

Use model as tool for studying individual differences

- **Architectural differences (e.g., visuospatial working memory ability)**
- **Knowledge-based differences**

Use model as tool for studying learning processes

- **Psychomotor and cognitive skill acquisition**
- **Following instructions**
- **Learning from instructions**